

# Pengolahan Citra Digital Dengan Visual Basic

*Artikel ini telah dipublikasikan pada majalah Mikrodata Vol. 8 Series 17, Agustus 2002.*

**Agus Kurniawan**

[agusk@cs.ui.ac.id](mailto:agusk@cs.ui.ac.id)

<http://blog.aguskurniawan.net>

## Pendahuluan

Pengolahan citra digital merupakan salah topik yang menarik baik kalangan mahasiswa, peneliti ataupun professional sehingga wajar kalau topik mengenai pengolahan citra digital atau lebih dikenal dengan Digital Image Processing dijadikan sebagai topik tersendiri yang berbeda dengan komputer grafik.

Pada artikel ini akan dibahas bagaimana kita melakukan pengolahan citra digital dengan memanfaatkan fungsi GDI API dan Visual Basic 6

## Pemrograman GDI

Untuk memanfaatkan GDI maka kita dapat menggunakan Win32 API untuk mengakses. Objek GDI itu sendiri tersimpan dalam file gdi32.dll. Berikut ini beberapa code untuk mengakses GDI API dengan Visual Basic

```
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long,
ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal
xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long
```

```
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As
Long) As Long
```

```
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function LoadImage Lib "user32" Alias "LoadImageA" (ByVal hInst As Long,
ByVal lpsz As String, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As
Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function GetTickCount Lib "kernel32" () As Long
```

Gambar yang akan digunakan dalam aplikasi yang akan dibuat bertipe bitmap 24 bit yang mempunyai struktur sebagai berikut

```
Private Type BITMAP
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type
```

Serta beberapa konstanta-konstanta yang membantu

```

Const IMAGE_BITMAP As Long = 0
Const LR_LOADFROMFILE As Long = &H10
Const LR_CREATEDIBSECTION As Long = &H2000
Const LR_DEFAULTCOLOR As Long = &H0
Const LR_COLOR As Long = &H2

```

## Menampilkan Gambar

Untuk menampilkan gambar dengan menggunakan GDI API maka kita dapat menggunakan fungsi BitBlt dengan memasukan data byte dari gambar tersebut

```

BitmapImage = GenerateDC(App.Path & "\Gambar.bmp", hbm)
If BitmapImage = Null Then
    Exit Sub
End If
GetObjectAPI hbm, Len(bm), bm

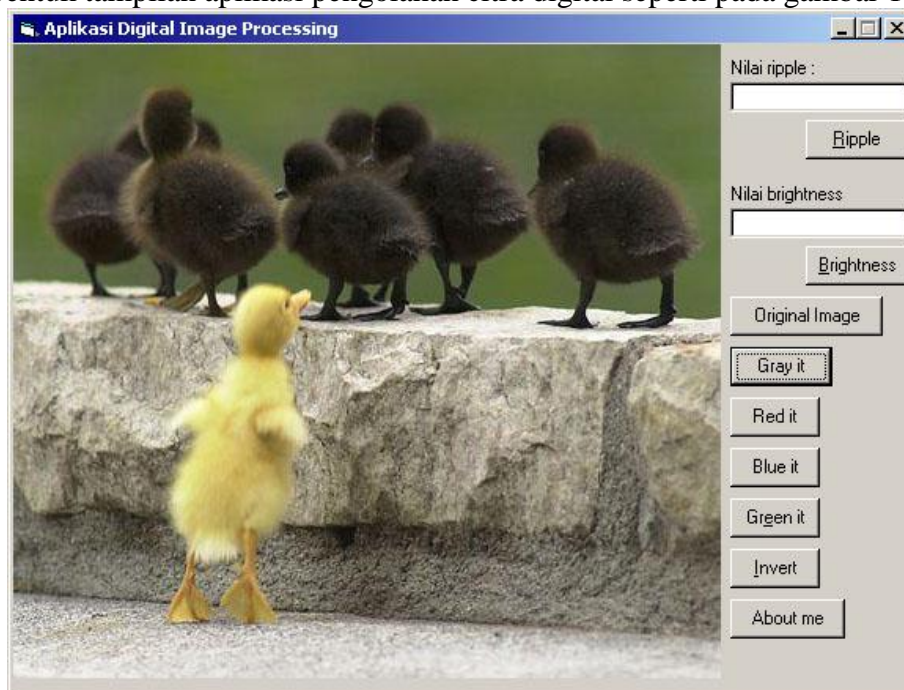
ReDim OriginalBits(1 To bm.bmWidthBytes, 1 To bm.bmHeight)

BitmapWidth = bm.bmWidth
BitmapHeight = bm.bmHeight

GetBitmapBits hbm, bm.bmWidthBytes * bm.bmHeight, OriginalBits(1, 1)
BitBlt Me.hdc, 0, 0, BitmapWidth, BitmapHeight, BitmapImage, 0, 0, vbSrcCopy

```

Contoh bentuk tampilan aplikasi pengolahan citra digital seperti pada gambar 1.



Gambar 1. Tampilan aplikai pengolahan citra digital

## Gambar Gray scale

Metode yang paling sederhana untuk mengubah gambar digital menjadi gambar gray scale yaitu menjumlah ketiga warna merah, biru dan hijau kemudian membaginya dengan nilai 3

```
For I = 1 To bm.bmWidthBytes Step 3
  For J = 1 To bm.bmHeight

    TempColor = OriginalBits(I, J)
    TempColor = TempColor + OriginalBits(I + 1, J)
    TempColor = TempColor + OriginalBits(I + 2, J)
    TempColor = TempColor / 3

    ByteArray(I, J) = TempColor
    ByteArray(I + 1, J) = TempColor
    ByteArray(I + 2, J) = TempColor

  Next J
Next I
```

Hasil running dari seperti pada gambar 3 dan gambar 2 adalah gambar aslinya



*Gambar 2. Gambar anak bebek*



*Gambar 3. Hasil proses gray scale*

## Filter Inverse

Metode yang sering dipakai adalah mengurangi nilai warna sebesar 255 terhadap warna yang dimiliki untuk semua warna. Berikut ini metodenya:

```
For I = 1 To bm.bmWidthBytes Step 3
  For J = 1 To bm.bmHeight

    ByteArray(I, J) = 255 - OriginalBits(I, J)
    ByteArray(I + 1, J) = 255 - OriginalBits(I + 1, J)
    ByteArray(I + 2, J) = 255 - OriginalBits(I + 2, J)

  Next J
Next I
```

Hasil filter inverse dapat dilihat pada gambar 4.



*Gambar 4. Hasil proses filter inverse*

## Filter Warna

Filter warna adalah filter untuk mengubah gambar ke arah warna tertentu atau bisa juga ke arah warna random. Cara yang dipakai yaitu setting kedua warna dari biru, hijau dan merah menjadi 0 sehingga tinggal satu warna yang akan mempengaruhi bentuk warna gambar. Misalkan untuk filter warna hijau:

```
For i = 1 To bm.bmWidthBytes Step 3
  For J = 1 To bm.bmHeight

    ByteArray(i, J) = 0
    ByteArray(i + 1, J) = OriginalBits(i + 1, J)
    ByteArray(i + 2, J) = 0

  Next J
Next i
```

Hasil dari filter warna hijau seperti terlihat pada gambar 5.



*Gambar 5. Hasil dari proses filter warna hijau*

Dengan cara yang sama, untuk filter warna biru sebagai berikut:

```
For i = 1 To bm.bmWidthBytes Step 3
  For J = 1 To bm.bmHeight

    ByteArray(i, J) = OriginalBits(i, J)
    ByteArray(i + 1, J) = 0
    ByteArray(i + 2, J) = 0
  Next J
Next I
```

Sehingga hasil dari filter warna biru seperti pada gambar 6 dan hasil filter warna merah juga dapat dilihat pada gambar 7.



*Gambar 6. Hasil dari filter warna biru*





*Gambar 7. Hasil dari filter warna merah*

## Ripple

Ripple itu sendiri merupakan salah efek gambar dengan memanfaatkan pola sinusoidal dimana masing-masing warna pada pixel dilakukan proses perhitungan sinusoidal. Berikut ini code untuk membuat efek ripple dengan fungsi Sin dan periodenya sebesar 1/5

```
For i = 1 To BitmapWidth
    TempValue = i + Sin(i / 5) * Val(txtRipple.Text)
    If TempValue > BitmapWidth Then
        RippleTable(i) = BitmapWidth
    ElseIf TempValue < 1 Then
        RippleTable(i) = 1
    Else
        RippleTable(i) = TempValue
    End If
Next i

ReDim ByteArray(1 To bm.bmWidthBytes, 1 To bm.bmHeight)

For i = 1 To bm.bmWidthBytes Step 3
    For J = 1 To bm.bmHeight

        ByteArray(i, J) = OriginalBits(i, RippleTable(J))
        ByteArray(i + 1, J) = OriginalBits(i + 1, RippleTable(J))
        ByteArray(i + 2, J) = OriginalBits(i + 2, RippleTable(J))
    Next J
Next i
```



*Gambar 8. Hasil dari efek ripple sebesar 5.*

## Brightness

Brightness juga merupakan salah satu efek gambar untuk membuat terang gambar sebesar nilai yang diinginkan. Metode yang sering dipakai adalah dengan mengalihkan semua nilai warna dengan nilai brightness yang diinginkan. Berikut ini contoh code-nya:

```
For i = 0 To 255
    TempValue = i * Val(txtBright.Text)

    If TempValue > 255 Then
        BrightTable(i) = 255
    Else
        BrightTable(i) = TempValue
    End If
Next i

ReDim ByteArray(1 To bm.bmWidthBytes, 1 To bm.bmHeight)

For i = 1 To bm.bmWidthBytes Step 3
    For J = 1 To bm.bmHeight

        ByteArray(i, J) = BrightTable(OriginalBits(i, J))
        ByteArray(i + 1, J) = BrightTable(OriginalBits(i + 1, J))
        ByteArray(i + 2, J) = BrightTable(OriginalBits(i + 2, J))

    Next J
Next i
```

Hasil dari efek brightness ini dapat dilihat pada gambar 9 dimana nilai brightness sebesar 5



*Gambar 9. Hasil dari efek brightness sebesar 5*

## **Referensi**

- MSDN library, "GDI Programming"